Foundation Of Robotics (Ludovic Righetti)

Shantanu Kumar (Sk9698)

1. Basics: using python and basic linear algebra I defined the following functions.
    - vec_to_skew(w) that transforms a 3D vector (numpy array) into a skew symmetric matrix.
    - twist_to_skew(V) that transforms a 6D twist into a 4x4 matrix (use vec_to_skew)
    - exp_twist_bracket(V) that returns the exponential of a (bracketed) twist e[V] where the input to the function is a 6D.
    - inverseT(T) that returns the inverse of a homogeneous transform T.
    - getAdjoint(T) that returns the adjoint of a homogeneous transform T.

2. Forward Kinematics: I have used product of exponential formula to calculate T, the new position of end effector frame. The formula: T(θ) = e[S1] θ1 · · · e[Sn−1] θn−1e[Sn]θnM

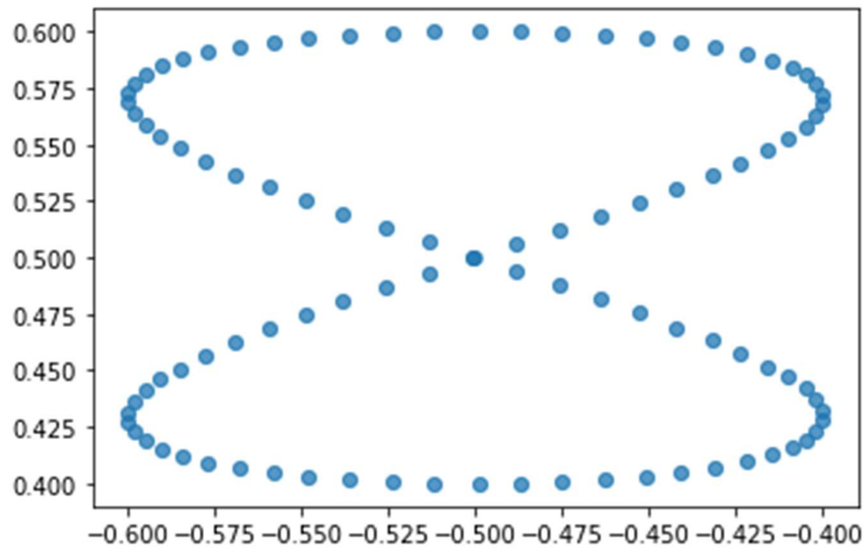
    - First, I tried to implement a function for a single value:
      T = (exp_twist_bracket(S1 * theta1)) * (exp_twist_bracket(S2 * theta2))*… ….*M
    - Then I tried implementing the loop for multiple values, I used a variable to calculate all the exponential values and then multiplied by M, the pose of end effector with respect to the base frame.
    - Created a function which took theta, pose and Screw and returned T(sb).

3. Jacobians: I used the formula to calculate ith column of J(θ):

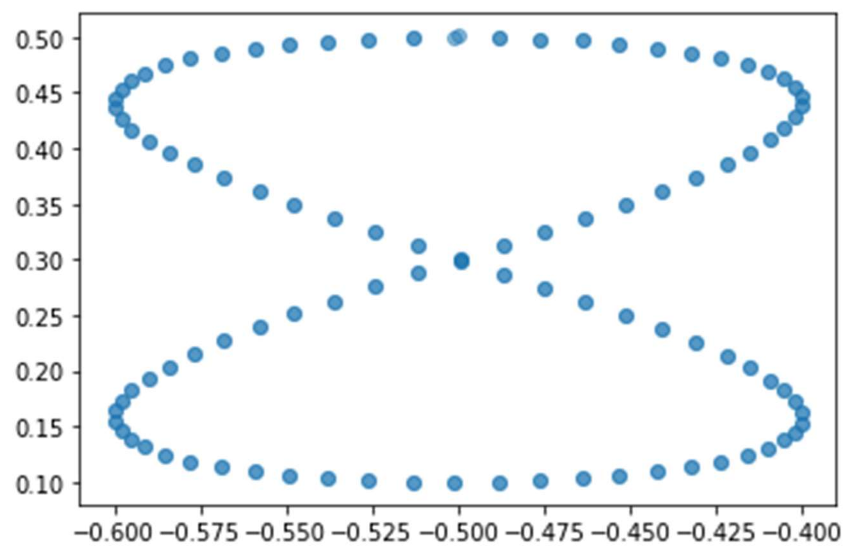
$$Ad_{e^{[s1]\theta 1} \dots e^{[si-1]\theta i-1}}(S_i)$$

   where $J_{S1} = S_1$ and S_i is the stack of skews containing linear and angular velocity.
    - Like forward kinematics, I first tried to solve for a single Jacobian value before implementing a loop.
    - Used a variable to calculate exponential values multiplied with theta for each jacobian value, took its adjoint and multiplied with screw.
    - I had to adjust the shape of Screw and Jacobian Column using np.reshape to use them in further calculations.
    - Created a function which took theta and screw and returned Jacobian Matrix.

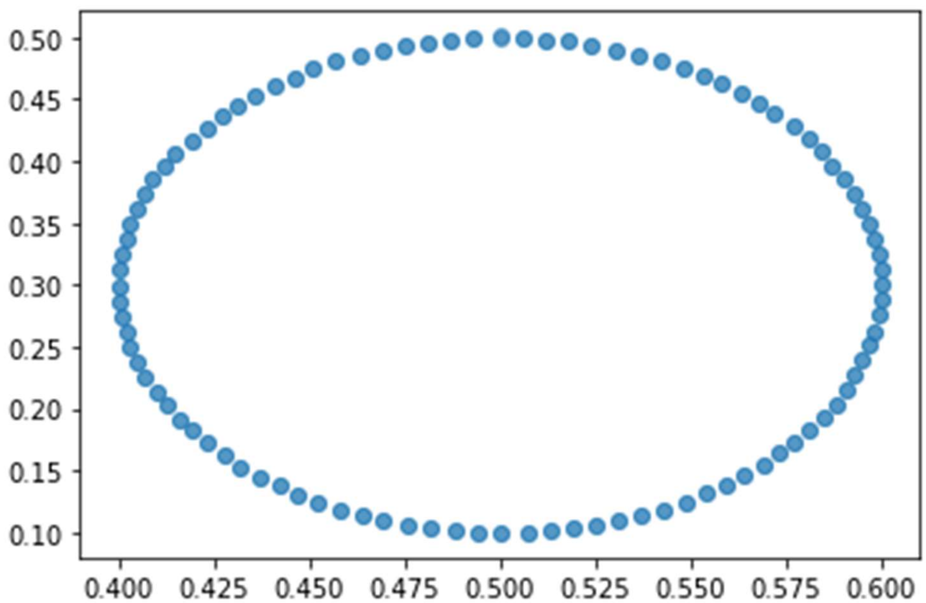
4. Displaying hand trajectory's
    - Imported values from given file 'Joint Trajectory.npy'
    - Created variables to store value of x,y and z.
    - Ran a loop for all the values of configuration and stored the value into variables.
    - Used Scatter plot to Store the Values:



X-Y plot

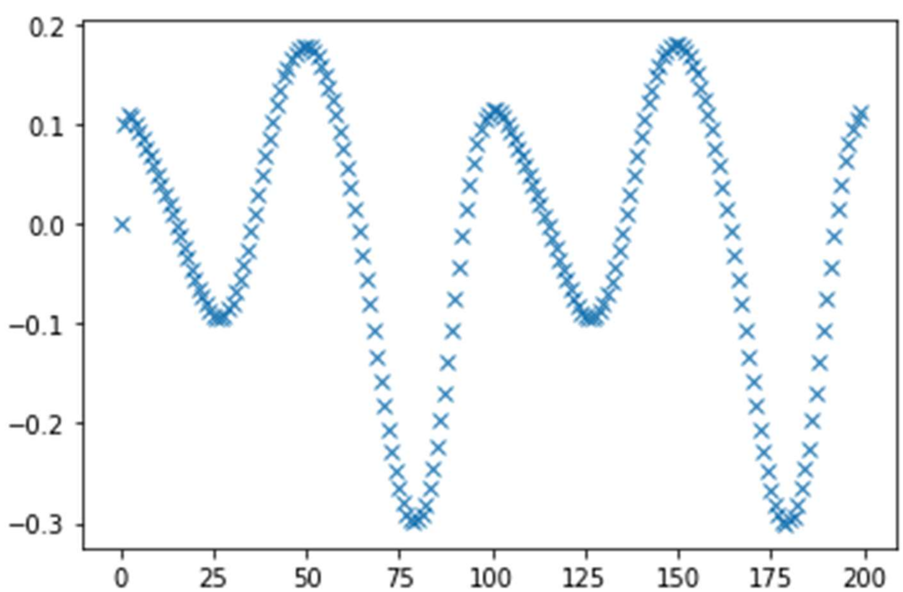


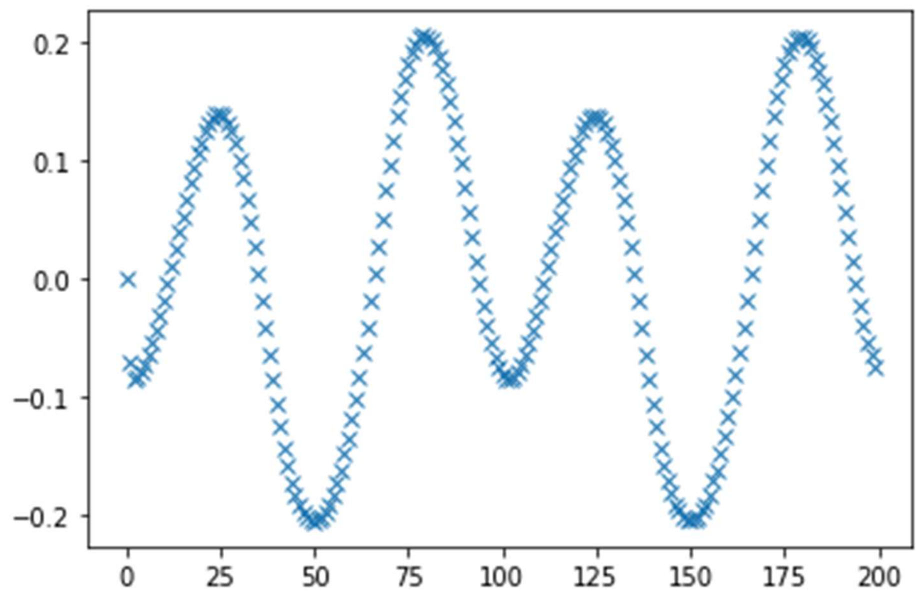
Y-Z plot



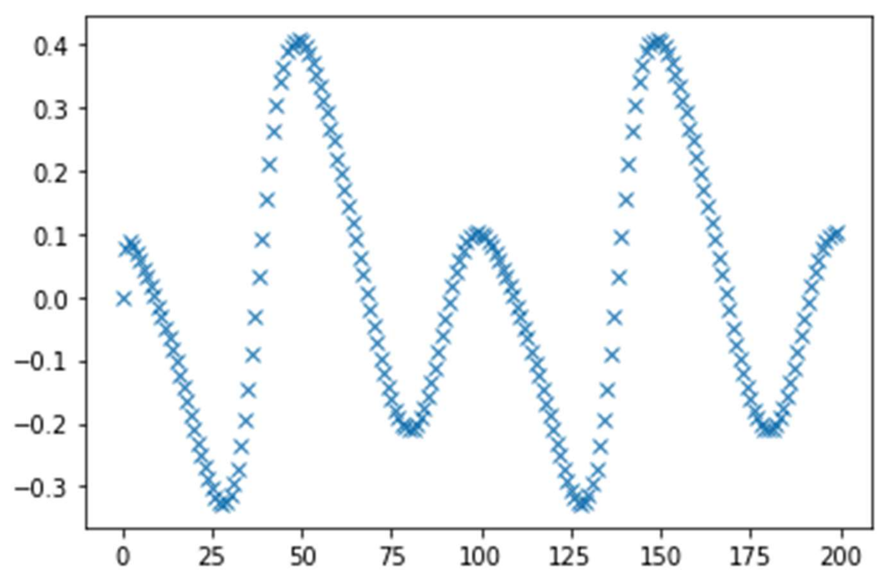
Y-Z plot

5. Computing Velocities: The formula used $V_s = J_s(\theta).\dot{\theta}$
    - Imported values from given file 'Joint Velocity.npy'
    - Declared variables to store values of x,y and z linear velocity
    - Made a loop to calculate Jacobian of all the configurations.
    - Multiplied the Jacobian to joint velocity to get end effector velocity in spatial frame. Used plt.plot; X-axis Plot:



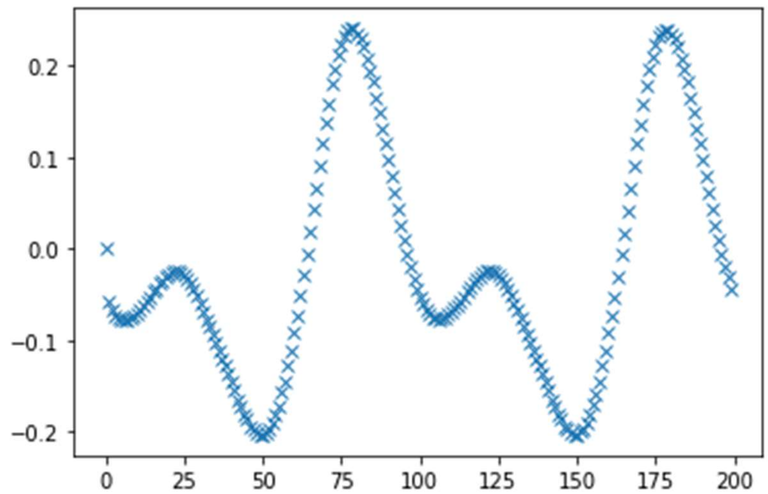


Y plot

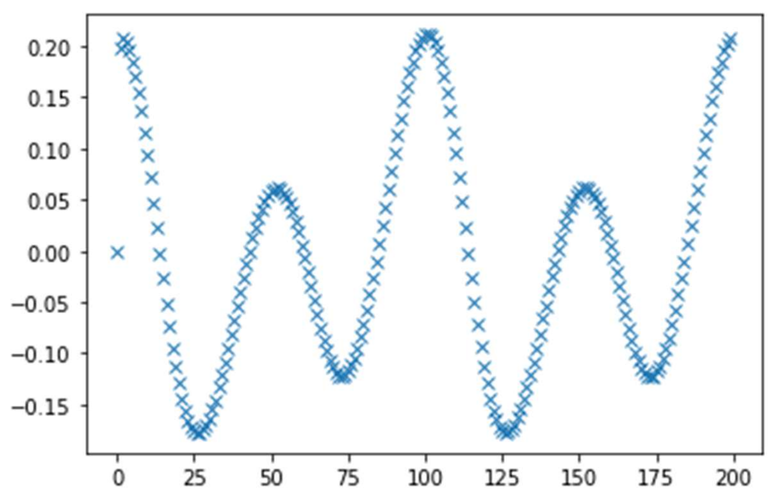


Z plot

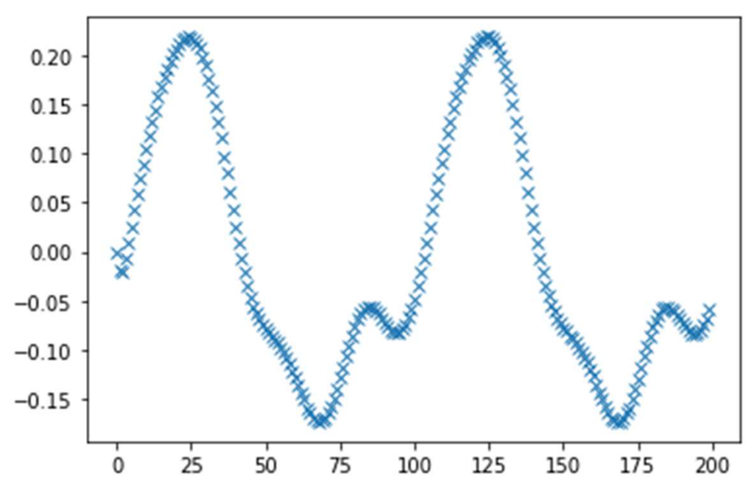
- To calculate bode frame velocity we need to calculate body Jacobian.
- Since, I had already calculated Spatial Jacobian, I decided to calculate the body Jacobian using adjoint of T(θ) : $J_{(b)}(\theta) = Ad_{T_{bs}}\left(J_{(s)}(\theta)\right) = \left[Ad_{T_{bs}}\right] J_{(s)}(\theta)$
- Took inverse of T(sb) to get T(bs) and calculated Adjoint.
- Used adjoint to convert spatial to body jacobian and calculated end effector velocity in body frame.



X-Plot

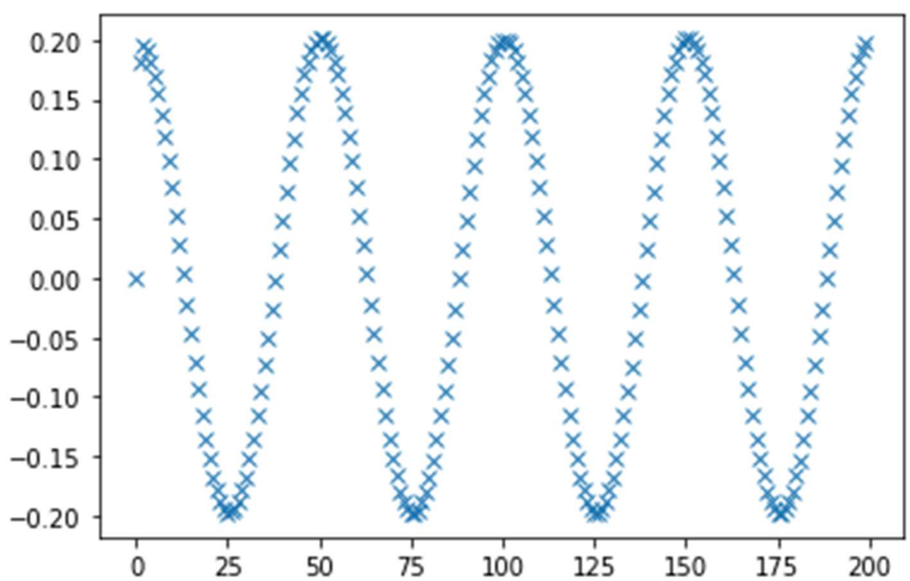


Y-Plot

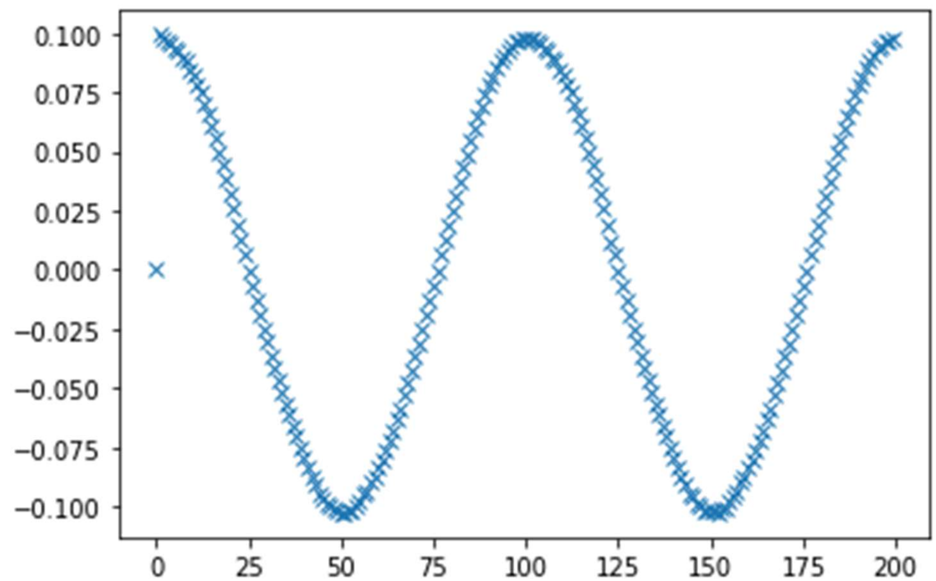


Z- Plot

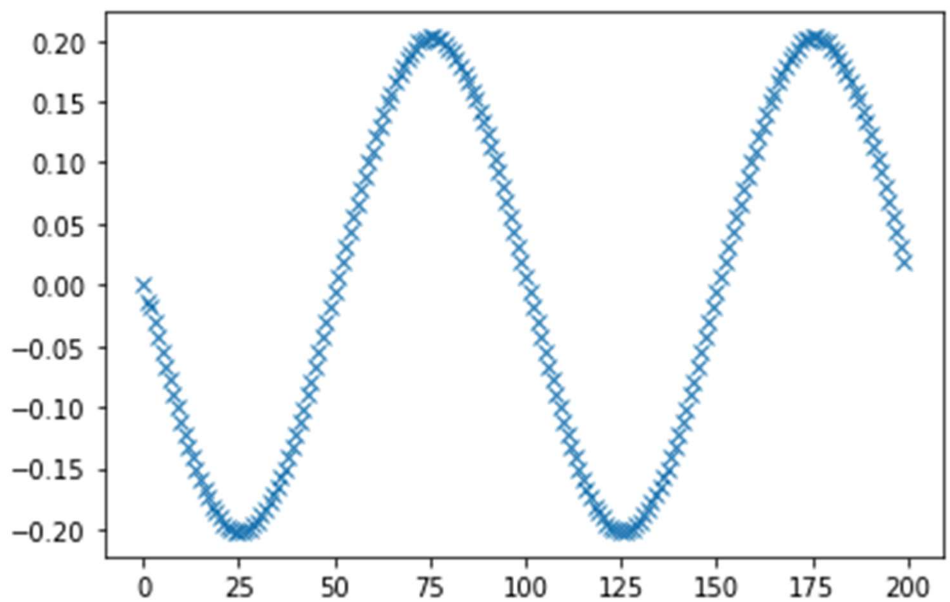
- To calculate the end effector frame oriented as base frame, I decided to remove rotation from the adjoint of T(bs) while converting the Spatial Jacobian to Body Jacobian.
- Removal of rotation matrix resulted in transition of frame from the spatial to end effector frame without changing the orientation.
- First, I calculated T(sb) using forward kinematics loop, took the co-ordinates of the position in a variable P.
- Stacked P with an Identity matrix to make a new homogeneous transform with coordinated of the end frame and orientation of base frame.
- Converted the spatial Jacobian to new Jacobian using the Adjoint transform.
- Calculated velocity of end-effector frame with new Jacobian.



X-Plot



Y plot



Z-plot

- According to me the third frame which consisted of Body oriented as spatial was most intuitive due to two reasons:
  a) It had no component of angular velocity present like in the spatial frame.
  b) There was no change in the perspective, due to movement of end effector frame like in the body frame.